作为一名 App 爱好者，如何正确地向开发者反馈问题？

涨汤　03月26日

注：本文作者是一名产品经理，负责设计过的产品有视频剪辑工具「猫饼」、魅族浏览器、Flyme 社区等。

我在之前的 Power+ 文章里曾经提到过，我和 MIUI 的结缘是从一个问题反馈得到答复开始的。当时我反馈了自己使用手机过程中的一个问题，并提出了自己的一点建议，**这个反馈得到了开发人员的肯定答复。**

> 受到这样的鼓励，我在之后的使用中，提交了成百上千个反馈给 MIUI 开发团队。这些反馈有的是软件的 BUG，有的是界面元素的问题，也有一些是对交互流程的建议。
>
> 这些建议当中，有超过三分之一被官方采纳，体现在 MIUI 的系统更新之中。**不仅我获得了体验更好的产品，还有许多其它用户也同样受益于此。就是这样一个正向的循环，让我体会到了很强的满足感。**

这段经历，让我对于「用户反馈」这个形式的沟通有着十分积极的态度。所以在后来参与到应用软件的运营、研发工作中时，我对于来自用户的反馈也会给予同样的重视。但是在工作的过程中，我观察到了两个现象：

1. 只有很小比例的用户，会直接联系开发者反馈问题或建

   议

2. 反馈建议或问题的过程中，真正提供有效信息的用户是少数

难道大家使用软件的过程中都不会遇到问题吗？并不是的，因为大部分用户在使用产品时遇到问题，选择的不是联系技术支持，而是在微博上发发牢骚。 这也不难理解，因为大多数的产品团队并没有做到及时友好地答复用户，无法鼓励用户在后续保持反馈的热情与积极性。 所以，对于体量比较大、人力资源充沛的团队来说，还要设置一些社交媒体客服。你可能在微博里提到了某个产品，他们的客服人员就会在下面回复。

还有一些用户，针对某个应用提出了很好的改进方案，同样也是发在社交网络中，虽然得到了其它用户的一致认可，但因为没有抄送给开发者，这个建议并没有产生实际的价值。

所以，**作为开发者，对于送上门的反馈，一定会当成宝贝。越是优秀的产品团队，越是如此。**少数派所分享、讨论的大都是优质应用，它们背后无论是团队开发者还是独立开发者，对于恰当的反馈一定都是欢迎而且重视的。

无论是作为一名产品经理，还是作为一名用户，我都希望有更多人通过反馈的形式，合理表达自己遇到的问题与建议，这样我们就能有更多体验良好的产品可以使用。

这篇文章想和大家分享一些有关「用户反馈」的建议，希望你在之后遇到产品的使用问题或是对产品有改进建议的时

候，能够参考文中的方法，直接联系到开发者，让问题得到解决。

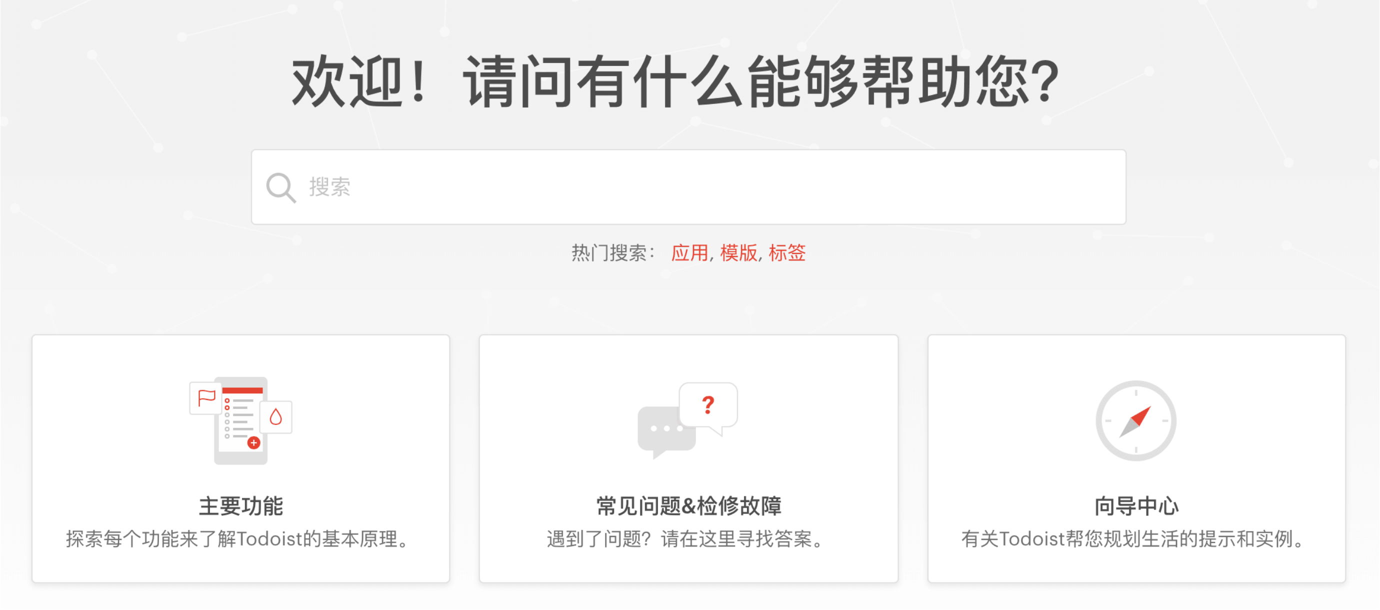
本文分为以下几个部分，可以根据自己的需要选择性阅读：

- 何时应该联系开发者?
- 如何有效地报告 BUG?
- 如何聪明地建言献策?
- 该如何联系到开发者?

何时应该联系开发者?

无论是团队还是独立开发者，他们的主要工作应该是致力于设计、研发新的产品功能，并且确保产品的质量。对于有一定使用门槛的产品，提供完善的用户手册也是他们应该做到的。

大多数情况下，你不应该直接向开发者询问产品的使用方法。除非你在用的产品还处于刚刚起步阶段，开发者还没有做好大部分的准备工作，例如准备基本的用户手册或者 FAQ 来回应常见问题。


欢迎！请问有什么能够帮助您?

热门搜索：应用, 模版, 标签

主要功能
探索每个功能来了解Todoist的基本原理。

常见问题&检修故障
遇到了问题？请在这里寻找答案。

向导中心
有关Todoist帮您规划生活的提示和实例。

Todoist 的用户支持页面

遇到使用问题的时候，**优先寻求用户手册、搜索引擎或者社群里的其它成员是正确的方式**，而且大概率能够解决你的问题。你要相信一点，**我们很难是第一个遇到某个问题的人。**

当你付出了一些尝试，仍然没有解决问题的时候，向开发者求助就成为你的首选了。除了描述你遇到的问题，不妨也说明一下你尝试解决问题的方法与过程。

有 3 种典型的场景，是比较适合直接联系开发者的：

1. 发现了产品的质量问题、功能缺陷，也就是 BUG
2. 产品无法满足你的特定需求，或者有值得改进的设计
3. 向开发者了解设计的细节，辅助你写测评文章等

这篇文章主要想讨论一下前面两种，也就是**报 BUG** 和**功能建议**，同时我会介绍一下和开发者联系的几种常见渠道与方式。

如何有效地报告 BUG?

BUG 在软件的世界里是永远存在的，作为使用者，我们一定都有遇到过 BUG。大部分人向开发者发起的第一次对话，也是报告一个 BUG。不过大部分人提交的报告对于开发者解决问题来说都是没有帮助的，因为开发者无法根据这些描述，在他自己的设备上重现（reproduce）你所遇到的问题。

很多 BUG 的定位，依赖于问题重现时产生的日志文件。所以想办法让开发者能够重现你所遇到的问题，对于修复 BUG 来说是至关重要的。 现在有越来越多的开发者工具，可以预置在你使用的应用里，当应用运行出现错误的时候，这些工具就会自动捕捉错误日志并上报给开发者。但是可以重现的问题场景，对于开发者判断和解决问题仍然十分重要。

对于报告 BUG 来说，有 4 点必须说明的内容：

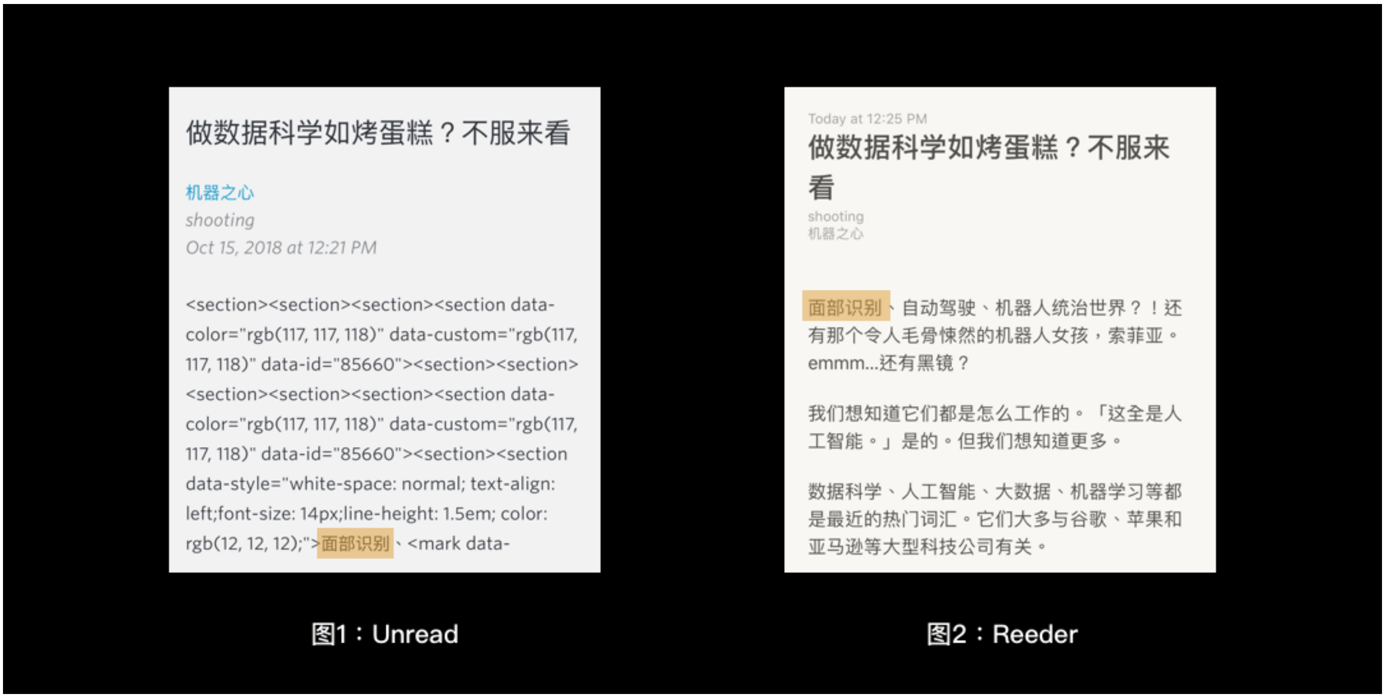
1. 重现问题的路径或操作
2. 期望得到的操作结果
3. 实际得到的操作结果
4. 问题发生的操作环境

例如我之前给 Unread一款 iOS 平台的 RSS 阅读应用的开发者邮件反馈过一个问题，描述如下（邮件原文是英文的，这里用中文重写了一次）：

- **操作路径**：
    - 在 Unread 打开来自 https://www.jiqizhixin.com/rss

    的文章

- **期望结果：**
    - 正确显示文章的内容（如下图2，截自另一款 RSS 客户端 Reeder）
- **实际结果：**
    - 文本内容包含了大量 HTML 标记，严重影响阅读（如下图1，Unread）
- **操作环境：**
    - Unread 1.9.7 (109064)
    - iOS 12.1 on iPhone



做数据科学如烤蛋糕？不服来看

机器之心

shooting

Oct 15, 2018 at 12:21 PM

```
<section><section><section><section data-color="rgb(117, 117, 118)" data-custom="rgb(117, 117, 118)" data-id="85660"><section><section><section><section><section><section data-color="rgb(117, 117, 118)" data-custom="rgb(117, 117, 118)" data-id="85660"><section><section data-style="white-space: normal; text-align: left;font-size: 14px;line-height: 1.5em; color: rgb(12, 12, 12);">面部识别、<mark data-
```

图1：Unread

Today at 12:25 PM

做数据科学如烤蛋糕？不服来看

shooting

机器之心

面部识别、自动驾驶、机器人统治世界？！还有那个令人毛骨悚然的机器人女孩，索菲亚。emmm...还有黑镜？

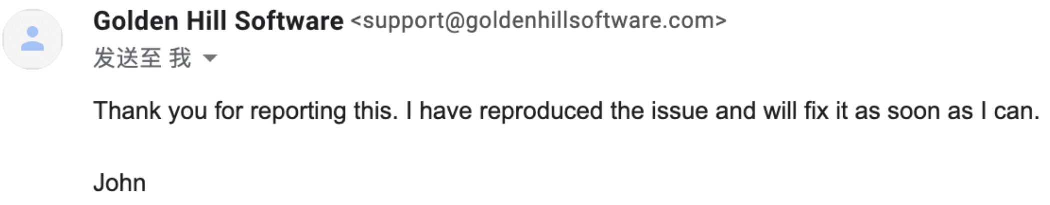
我们想知道它们都是怎么工作的。「这全是人工智能。」是的。但我们想知道更多。

数据科学、人工智能、大数据、机器学习等都是最近的热门词汇。它们大多与谷歌、苹果和亚马逊等大型科技公司有关。

图2：Reeder

给 Unread 报 BUG 时的附图

当天我就收到了开发者 John 的回信：



Golden Hill Software <support@goldenhillsoftware.com>

发送至 我

Thank you for reporting this. I have reproduced the issue and will fix it as soon as I can.

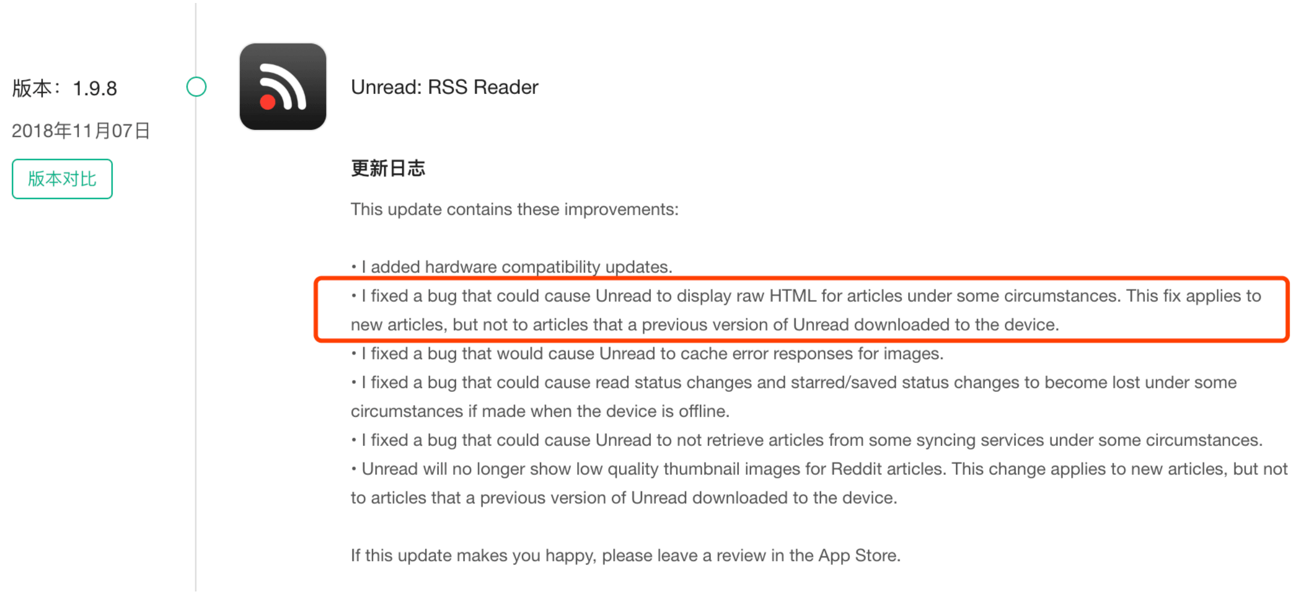
John

Unread 开发者给我的回信

注意看，John 在回信中提到他成功重现了我所说的问题：

> Thank you for reporting this. **I have reproduced the issue** and will fix it as soon as I can.

在 Unread 随后的一次更新当中，我反馈的这个问题就得到了解决。



版本：1.9.8

2018年11月07日

Unread: RSS Reader

更新日志

This update contains these improvements:

- I added hardware compatibility updates.
- I fixed a bug that could cause Unread to display raw HTML for articles under some circumstances. This fix applies to new articles, but not to articles that a previous version of Unread downloaded to the device.
- I fixed a bug that would cause Unread to cache error responses for images.
- I fixed a bug that could cause read status changes and starred/saved status changes to become lost under some circumstances if made when the device is offline.
- I fixed a bug that could cause Unread to not retrieve articles from some syncing services under some circumstances.
- Unread will no longer show low quality thumbnail images for Reddit articles. This change applies to new articles, but not to articles that a previous version of Unread downloaded to the device.

If this update makes you happy, please leave a review in the App Store.

Unread 的更新日志

1. 重现问题的路径或操作

在你遇到 BUG 之前，你一定做过某些操作，例如点击了某个功能按钮；如果是一连串的操作，就称为操作路径，在访问了特定功能页面以后，再点击某个按钮。

所以在你遇到 BUG 以后，不妨再重复一次之前的操作，看看问题是否还会发生。如果是的话，尽可能完整的描述你做过的所有操作。不要担心描述得太多，开发者可以从中找到

有用的部分。

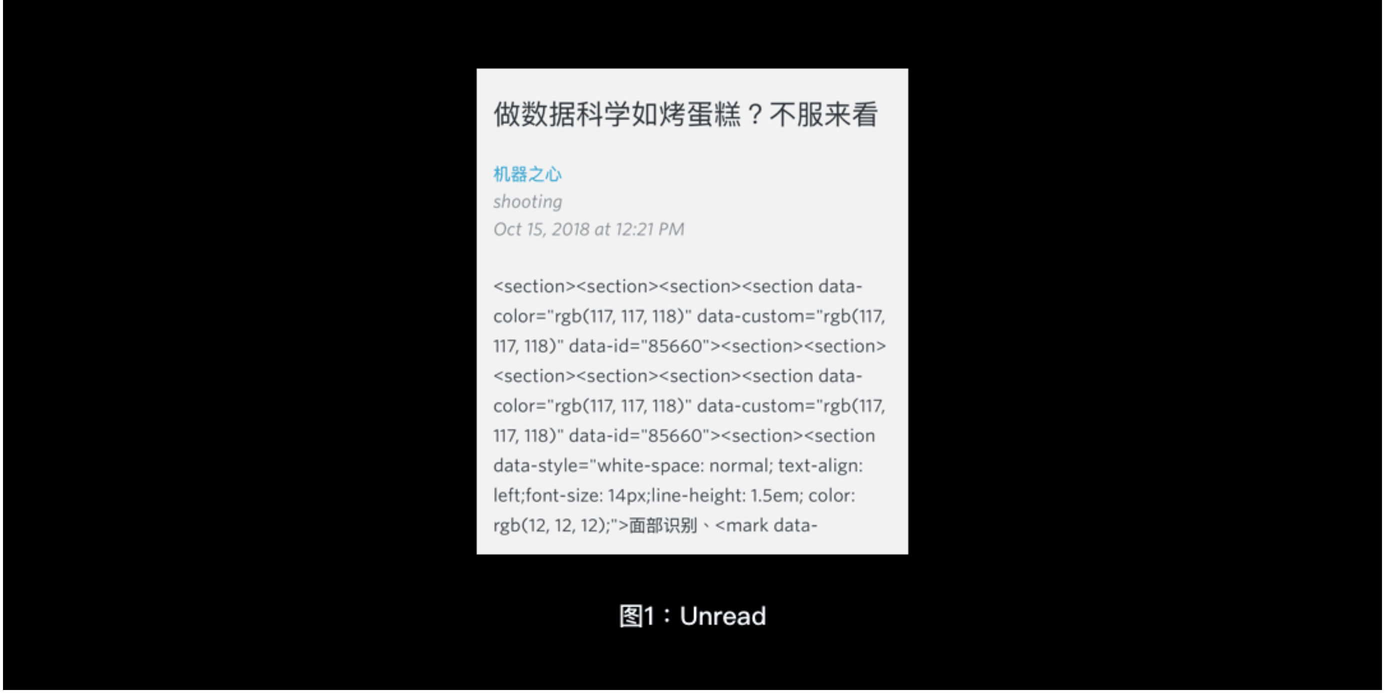
主流的操作系统都内置了屏幕录像功能，如果能够用录制你操作过程的视频，会更加有帮助。视频完整重现了整个过程，你甚至不需要用任何文字描述。

在上面的例子中，尽管**操作路径**只写了一个步骤，但是我指明了遇到问题的内容来源都是「机器之心」这个网站。所以开发者只需要在他的设备上同样尝试这个内容来源就能验证我提交的问题了。

2. 期望得到的操作结果

大部分情况下，在应用中执行固定的操作会得到固定的结果，这就是程序。就像上面的例子当中，打开阅读器以后，期望得到的是一篇篇正常显示的文章，似乎不言自明。但是我仍然说明了这一点，并且还配上了同一篇文章在另一个 RSS 阅读器里正常显示的截图。

这样做并不是多余的，因为遇到问题的人，和开发者所处的使用环境可能是不一样的。还是以上面的图为例，如果我仅仅提供了出问题的截图，那么开发者会不会以为这是一段 HTML 的示例代码？尤其是他还不一定认得中文。



做数据科学如烤蛋糕？不服来看

机器之心

shooting

Oct 15, 2018 at 12:21 PM

```
<section><section><section><section data-
color="rgb(117, 117, 118)" data-custom="rgb(117,
117, 118)" data-id="85660"><section><section>
<section><section><section><section data-
color="rgb(117, 117, 118)" data-custom="rgb(117,
117, 118)" data-id="85660"><section><section
data-style="white-space: normal; text-align:
left;font-size: 14px;line-height: 1.5em; color:
rgb(12, 12, 12);">面部识别、<mark data-
```

图1：Unread

Unread 的问题表现

如果他不觉得这是个问题，也就无从解决了。即便我们认为是显而易见的事情，也尽量附上说明，让看到反馈的开发者获得完整的了解。

3. 实际得到的操作结果

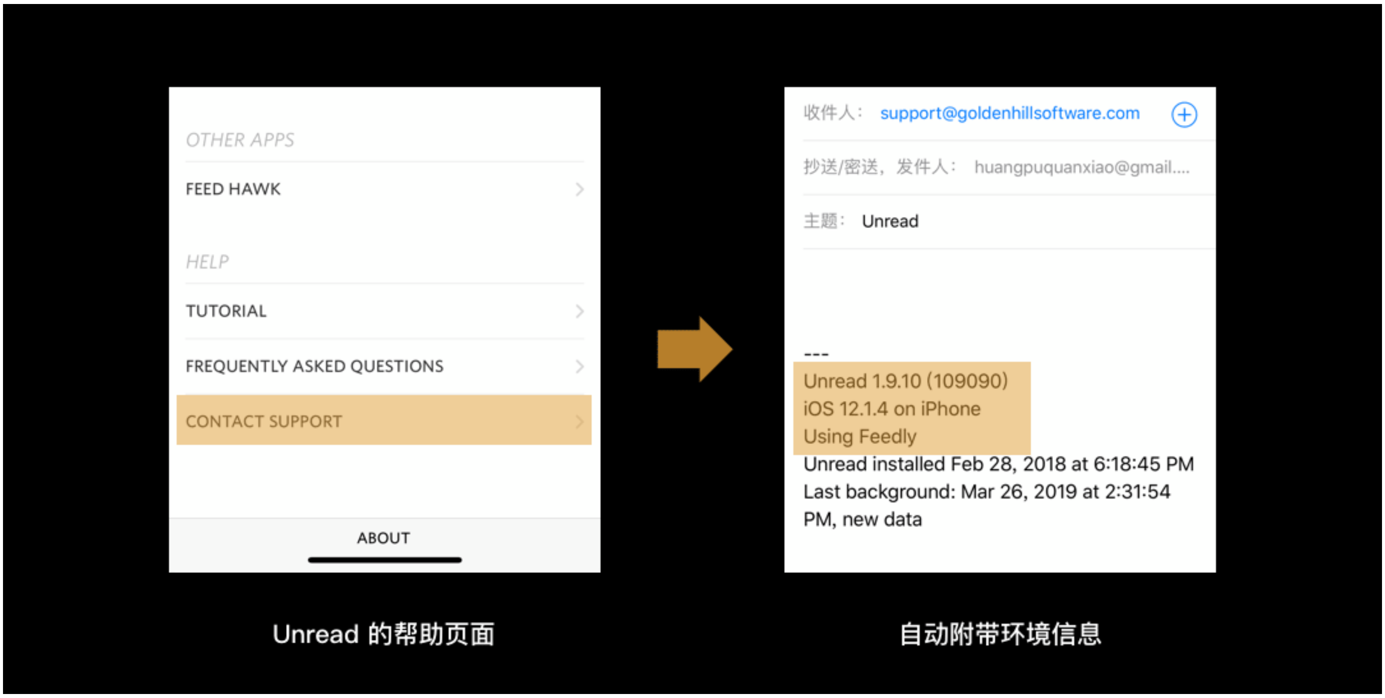
当然，对问题本身的描述必不可少，需要描述的就是我们看到的操作结果。反馈时，尽量配上问题出现的录屏或者截图，连截图都没有的时候，才仅用文字来详细描述。由于大家的表述习惯不同，甚至是语言不同，单纯用文字描述，不一定能够准确地达意。

4. 问题发生的操作环境

有些 BUG 只在特定的操作环境下能够重现，所以说明操作环境也是必须的。操作环境包括**软件环境**和**硬件环境**：

- **软件环境**包括你使用的应用软件和操作系统的名称及版本号，如：Unread 1.9.7
- **硬件环境**包括你使用的设备型号以及必要的配置信息，如：iPhone XS

如果你使用的是应用开发者提供的反馈通道，例如应用内的 BUG 反馈入口，这些信息往往已经自动附带，不需要额外填写。


OTHER APPS
FEED HAWK
HELP
TUTORIAL
FREQUENTLY ASKED QUESTIONS
CONTACT SUPPORT
ABOUT
Unread 的帮助页面
自动附带环境信息

收件人： support@goldenhillsoftware.com

抄送/密送，发件人： huangpuquanxiao@gmail....

主题： Unread

```
---
Unread 1.9.10 (109090)
iOS 12.1.4 on iPhone
Using Feedly
Unread installed Feb 28, 2018 at 6:18:45 PM
Last background: Mar 26, 2019 at 2:31:54
PM, new data
```

Unread 的邮件反馈渠道

有些情况下，也许还需要一些额外的信息才能帮助开发者定位到问题，尽可以等开发者回信来询问时再提供。**上述信息已经能展示我们是一名极其专业的用户了，开发者一定不会放过这个解决问题的机会的。**

如何聪明地建言献策?

上面说到的 BUG 属于应用软件的质量问题，就像是地板有

块砖松动了似的，如果这块松动的砖处在人来人往的要道上，那就很容易引起重视，得到修复。如果你觉得一间屋子的格局不合理，希望调整一下，或者想往这栋楼里添置点家具或电器，这就像是给开发者提建议，让他为你添加想要的功能或者特性，需要明确说明自己的问题与需求.

这个部分想讨论的是以下几点：

1. 无法评估的建议
2. 详细描述试图解决的问题
3. 开发者最爱听故事

1. 无法评估的建议

为数不少的用户，通过反馈渠道给我的产品提建议，或是希望增加某个新功能的时候，往往就几个字，例如：

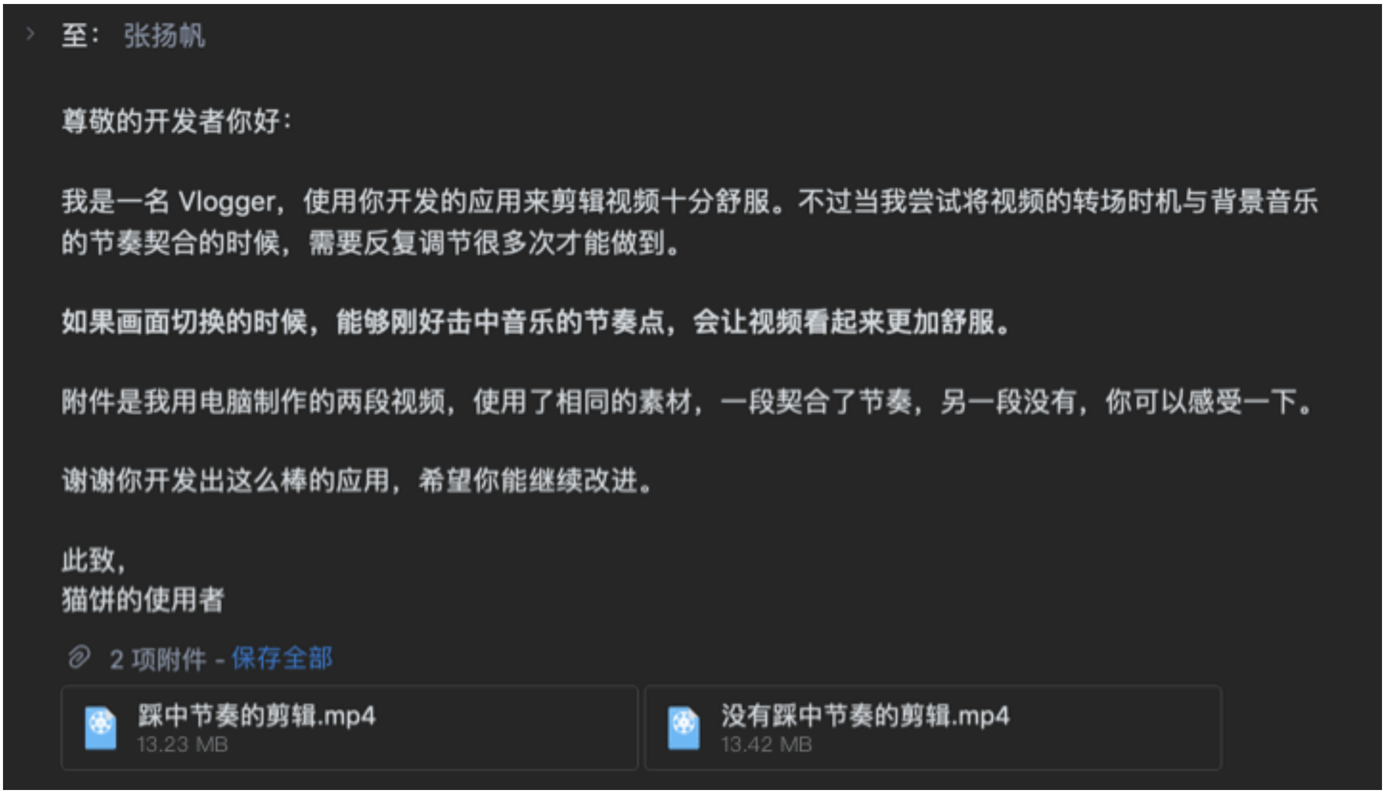
> 夜间模式
>
> 希望视频能自动分段
>
> 想要语音功能

你知道这些分别指的是什么吗？结合业务场景，我倒是能理解这些建议指向的都是什么功能，当然也有些时候是理解不了的。所以，首先要保证你的表达能够让开发者理解。**更大的问题在于，身为开发者，为什么我应该实现某个功能给你来使用呢？**

一个建议如果想被采纳，必须让开发者觉得你所提出的要求是合理的。所以，当你产生一个念头「这个应用要是能增加 XX 功能就好了」的时候，不能直接把这句话发给开发者，你需要给他充分的理由，让他理解你的问题，而不仅仅是告诉他应该增加某个功能。

2. 详细描述试图解决的问题

让开发者理解你的情况，最好的办法是完整地讲述你的问题或需求，由他来思考是否能够替你解决问题，以及应该怎么解决。在我设计视频编辑工具的时候，曾经收到过这样的反馈：



至： 张扬帆

尊敬的开发者你好：

我是一名 Vlogger，使用你开发的应用来剪辑视频十分舒服。不过当我尝试将视频的转场时机与背景音乐的节奏契合的时候，需要反复调节很多次才能做到。

如果画面切换的时候，能够刚好击中音乐的节奏点，会让视频看起来更加舒服。

附件是我用电脑制作的两段视频，使用了相同的素材，一段契合了节奏，另一段没有，你可以感受一下。

谢谢你开发出这么棒的应用，希望你能继续改进。

此致，
猫饼的使用者

2 项附件 - 保存全部

踩中节奏的剪辑.mp4
13.23 MB

没有踩中节奏的剪辑.mp4
13.42 MB

用户给猫饼的功能建议

这名用户的反馈非常打动我，因为他不仅没有指手画脚地告诉我应该给软件增加什么功能，而且和我分享了他让视频变

得更好看的心得。通过他提供的对比视频，我能明显感受到差异。

于是我通过邮件和这位用户进一步请教了「音乐契合节奏」这个技巧的更多信息，最终设计出了很好用的节奏打点剪辑功能。

所以，在提建议的时候，与其一味的告诉开发者应该增加什么功能，不如花心思把自己的故事讲好。

3. 开发者最爱听故事

「用户故事」是一些开发者用来描述功能的方式，从用户的角度来描述用户渴望得到的功能。一个好的用户故事包括三个要素：

1. 角色，什么样的用户需要使用这个功能
2. 活动，具体要做的事情是什么
3. 价值，达成这件事情对于用户、对于产品来说，有什么好处

以上面的用户建议为例，我最终把功能写成这样一条用户故事：

> 作为一名 Vlogger，我希望能够标记音乐的节奏点，这样我在剪辑的时候，可以让视频转场时，与音乐节奏结合的更好，从而让视频更加好看。

当开发者写出用户故事的时候，就很容易判断这个功能是否

值得开发出来。所以，我们想要自己的创意被开发者实现，不如多讲讲自己的「故事」。如果你的故事打动了开发者，那么开发者自然也会来询问你是否有解决问题的建议。

再次提醒，单纯反馈说应用「不好用」，或者以命令的口吻要求开发者为你增加某个功能，都不可能得到让你满意的结果。

一般来说，开发者对自己的应用都有一定的规划，我们通常称为产品路线图。有些产品团队甚至会把路线图公示出来，例如 OmniGroup 会在每年初的时候，公开他们当年的产品路线图（Roadmap）。这就意味着，你提出的建议，优先级必须大于掉路线图上的那些待开发需求，才有可能被尽快实现出来。

该如何联系到开发者?

当我们需要和开发者沟通的时候，有很多可能的途径能触达他们。如果你比较在意自己能否收到反馈，可以先判断一下开发者接受反馈的意愿如何，判断的方式就是找到他尽可能多的联系方式：

- 电子邮件地址
- 用户反馈功能
- 社交网络账号
- 社群或论坛
- 应用商店评论

开发者如果在社群、论坛、社交网络比较活跃，你能看到他在答复其它用户，那么你提交的反馈得到答复的机会也会很大。

电子邮件

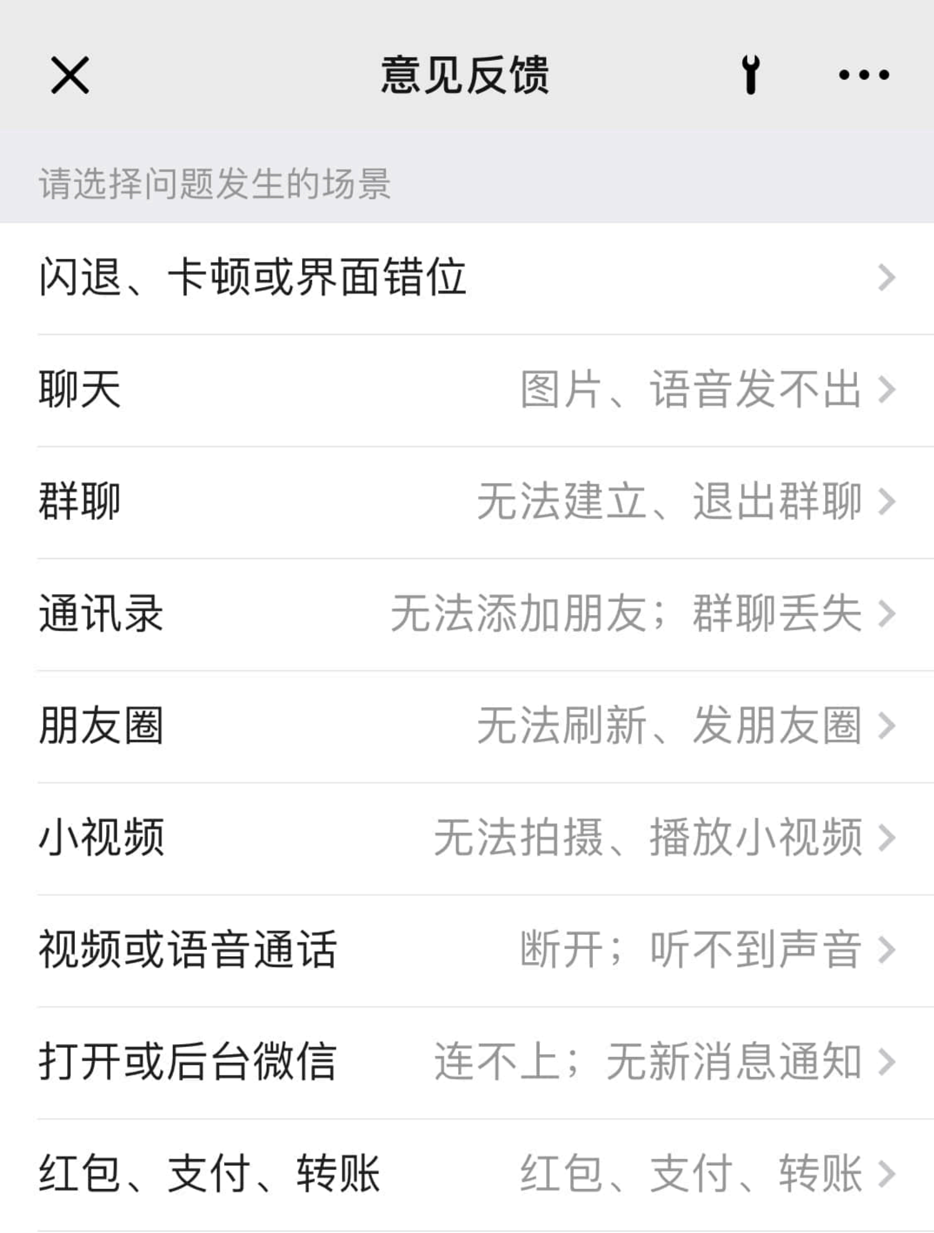
我最推荐的反馈方式是电子邮件，邮件虽然「古老」，但是十分适合：

- 邮件这个形式本身比较正式，适合反馈这个场景
- 邮件的附件可以包含所有类型的文件，包括媒体和日志
- 开发者接收反馈通常是专门的邮箱，不容易错过

如果应用内的反馈功能指向的是一个电子邮件地址，十分推荐使用，因为它还替你附带了必要的操作环境信息和一些你使用当前应用的日志，开发者会更加了解你的情况。如果应用内没有入口的话，从应用商店的应用信息或者产品的官网通常也能找到。

用户反馈

开发者在应用内提供反馈的入口，一般来说是积极的信号。之所以不作为第一选择，是因为我很少看到中小型团队接入用户反馈功能，并且还能维护得很好的。对于大公司的产品来说，针对反馈的答复往往比较格式化，很难让人感到满意。



意见反馈

请选择问题发生的场景

闪退、卡顿或界面错位	
聊天	图片、语音发不出
群聊	无法建立、退出群聊
通讯录	无法添加朋友；群聊丢失
朋友圈	无法刷新、发朋友圈
小视频	无法拍摄、播放小视频
视频或语音通话	断开；听不到声音
打开或后台微信	连不上；无新消息通知
红包、支付、转账	红包、支付、转账

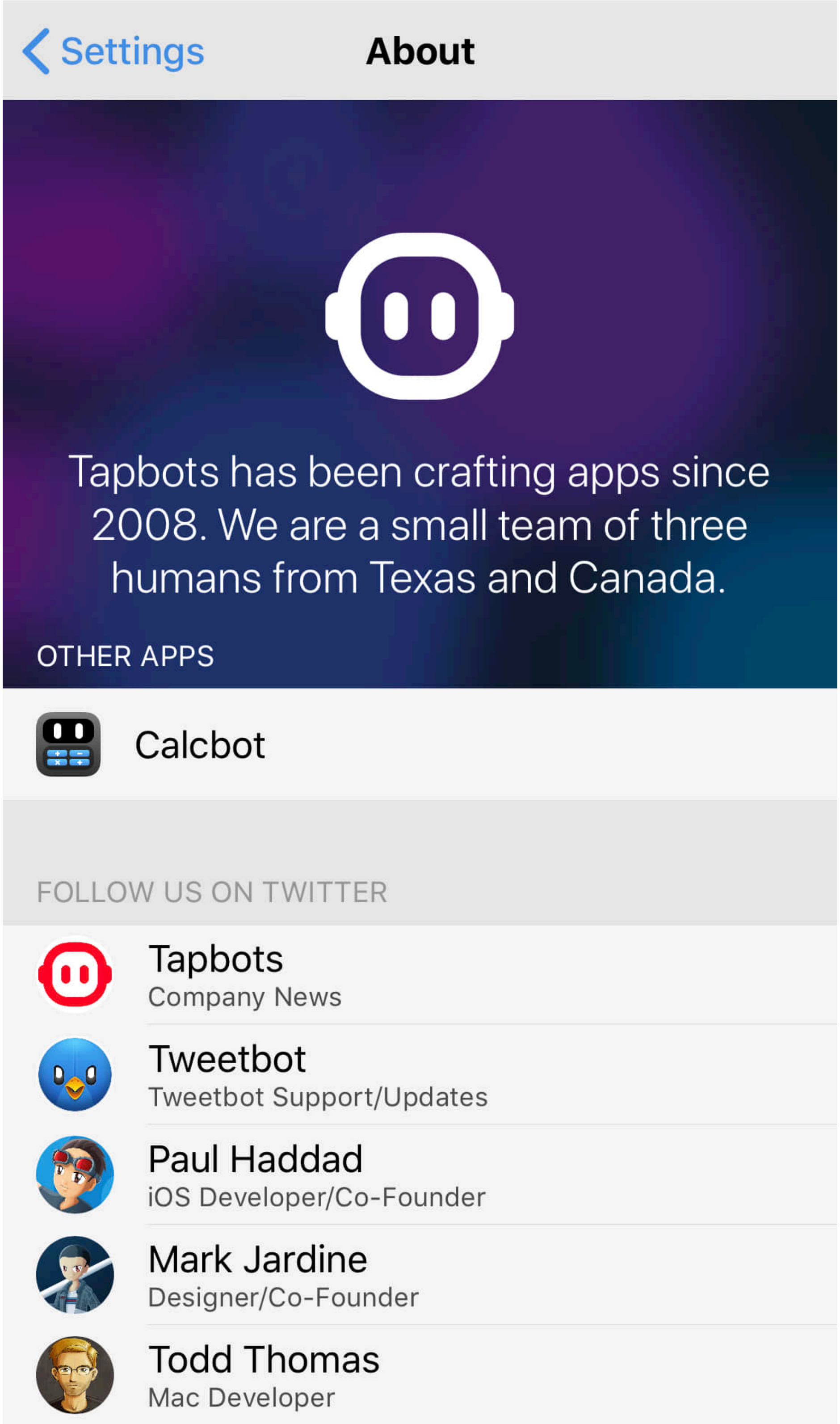
微信的意见反馈页面

不过在应用内提交个反馈并不麻烦，所以遇到问题时，不妨先试一下。如果没有得到理想的答复，可以再尝试发一封邮件过去。

社交网络

微博、推特或者其它开发者会出没的社交网络也是不错的渠道，你可以把问题或建议的描述写成一条贴文，然后 @ 一下开发者，让他看到。也可以通过私信的方式发给他。

@ 的好处在于，如果开发者答复你的问题，同时也可以让其它使用者看到。如果在推特或者论坛里，这些内容还能够被搜索引擎索引，方便更多人。



Tapbots has been crafting apps since 2008. We are a small team of three humans from Texas and Canada.

OTHER APPS

Calcbot

FOLLOW US ON TWITTER

Tapbots
Company News

Tweetbot
Tweetbot Support/Updates

Paul Haddad
iOS Developer/Co-Founder

Mark Jardine
Designer/Co-Founder

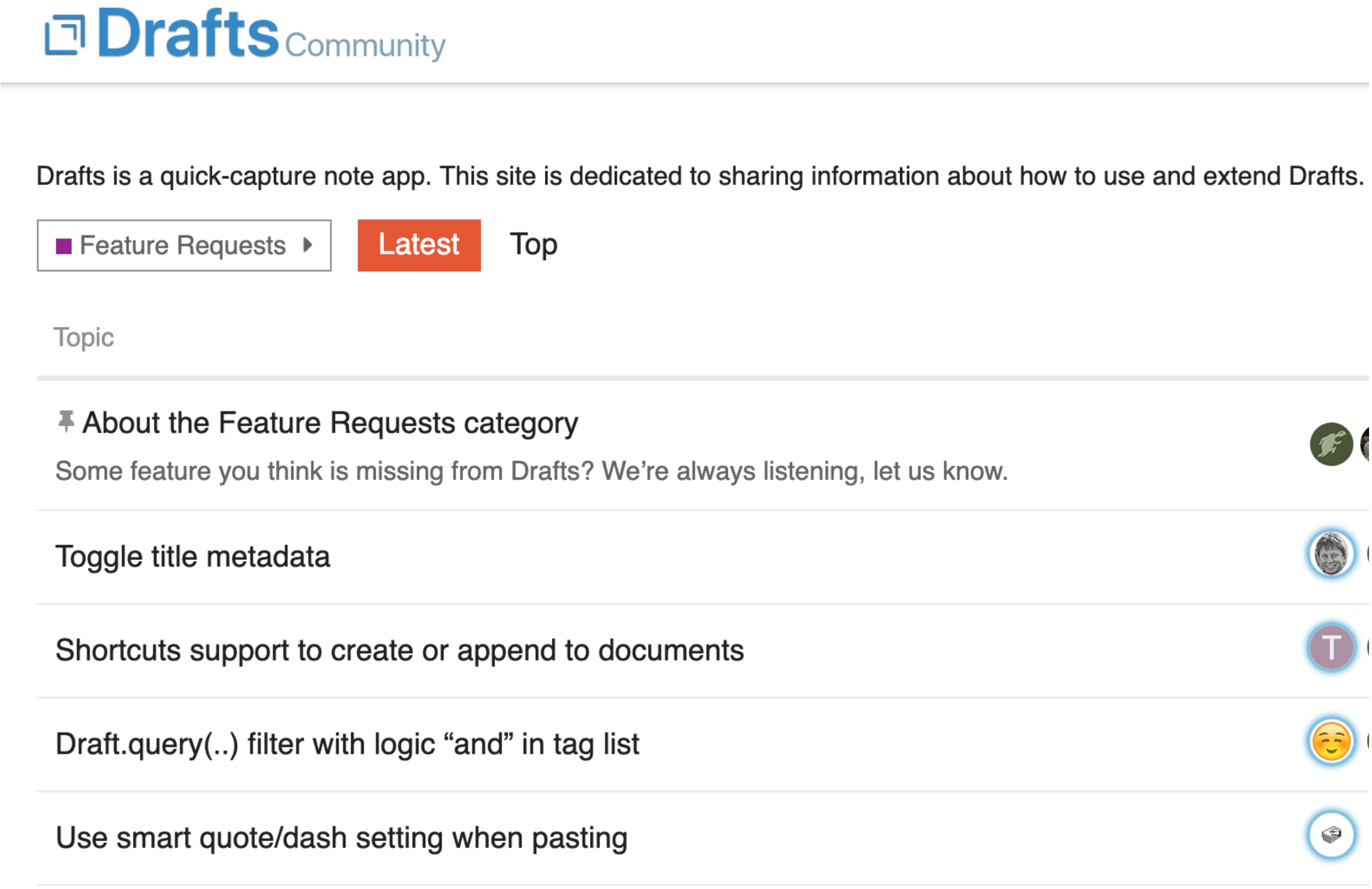
Todd Thomas
Mac Developer

Tweetbot 的关于页面

有很多独立开发者经常在社交网络上活跃，他们会把自己的账号展示在应用的设置或者关于页面上，很容易就能找到。

社群或论坛

有些开发者会搭建论坛，或者组织 Slack 群组的方式，将他的用户聚集在一起，这样既能促进交流，也能及时获得各种反馈。Power+ 的读者群也是这样的存在，Power+ 的产品是文章，读者们就是用户，可以提出各种问题和建议。



Drafts is a quick-capture note app. This site is dedicated to sharing information about how to use and extend Drafts.

About the Feature Requests category

Some feature you think is missing from Drafts? We’re always listening, let us know.

Toggle title metadata

Shortcuts support to create or append to documents

Draft.query(..) filter with logic “and” in tag list

Use smart quote/dash setting when pasting

Drafts 官方论坛的功能建议板块

如果开发者有心维护这样的社群，你对他能够友善回应你的反馈可以很有信心。

商店评论

如果上述联系方式你都没有找到，还可以试试在应用商店提交一条评价（Review），将你想要反馈的问题或是建议写在这里。如果这个应用的评价数量不算太多，那么开发者还是很有机会看到你的留言的。

目前主流的应用商店都有「开发者回复」这一功能了，所以一样能够获得答复。

总结

上面的建议都属于经验之谈，当你和特定的开发者沟通的时候，总还有其它要注意的事情。例如反馈 BUG 的格式，很多产品其实会提供模板，你就照着模板来填写最为理想。

如果一个用户能够做到本文说的这些注意事项，至少我作为开发者已经十分感激了：

- 提交 BUG 反馈的时候提供重现问题的步骤
- 提交新功能建议的时候着重讲解自己的需求
- 尽可能直接联系到开发者

另外还有一点很重要，那就是礼貌，无论你说的哪种语言，在和陌生人打交道的时候，用一些敬语不是坏事。如果修辞掌握的不是很好，用简单、直接的文字表述问题就好。

也确实有开发者不够重视用户的反馈，但如果你已经做到了我说的这些，不必因为某些开发者没有答复你的反馈，就丧失了反馈的热情。

祝你提交的 BUG 都被很快修复，提出的建议都被开发者变为功能：）

注 1：@文刀漢三 与 @咕毛 对本文亦有贡献。
注 2：本文题图由@咕毛 设计。

参考阅读

人人都应该阅读用户反馈邮件

我对开发者与用户通过「反馈或建议」建立的联系十分认同，主要是自己受益颇多。想写这个选题主要是被这篇文章所提醒。如果你的工作是服务于一批用户的，建议你读一读这篇文章。

这篇文章讨论了身为开发者，或者产品团队的一员，为什么应该重视用户反馈。不仅仅是因为用户们提供了 BUG 的反馈或者是功能的建议。

更重要的是，这是人与人交流的过程，用户的反馈里往往还包含了情绪和情感，传达的可能是肯定也可能是失望，但这些都会让你感觉到自己所做的产品是有价值的，也会更加有责任感。

🔗：Why Everyone Should Read Customer Support Emails

提问的智慧

本文原文由知名 Hacker Eric S. Raymond 所撰写，教你如何

正确地提出技术问题并获得你满意的答案。

这篇文章同样比较长，而且例子是偏技术向的问题，不过抛开技术的部分，其中的提问方式同样适用于我们在其它场合下使用。

事无巨细的讲解了提问前、提问中、提问后要注意的事情，包括了如何描述问题、注意文本的内容与格式、表述问题时的礼节等等。

🔗：提问的智慧

英文原文：How To Ask Questions The Smart Way

如何有效地报告 Bug

这篇文章由知名程序员 Simon Tatham 所写，他主要因创建和维护 PuTTY 而闻名。许多软件开发人员推荐用户在向他们报告软件错误前阅读他的这篇文章。

尽管文章中的举例是基于 PC 时代的电脑软件，但是其中的要点仍然没有过时，也就是我在文中提到的，尽可能帮助开发者重现问题，这样才能更快的定位并解决问题。

文章对于一些提问时的常见错误也有列举，如果你的提问总是没有得到答复，可能是犯了某些你没意识到的问题，不妨看看这篇文章对号入座一下：）

🔗：How to Report Bugs Effectively